

AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

- 1 1. (Previously Presented) A method of using a virtual machine monitor and an operating  
2 system on computer hardware in a computer, the method comprising:  
3 interposing the virtual machine monitor between the computer hardware and the  
4 operating system at runtime, wherein the interposing occurs after booting of the computer, and  
5 wherein interposing the virtual machine monitor gives the virtual machine monitor direct control  
6 of at least a portion of the computer hardware; and  
7 booting the operating system on the computer hardware before interposing the virtual  
8 machine monitor at runtime.
- 1 2. (Cancelled)
- 1 3. (Previously Presented) The method of claim 1, further comprising booting the virtual  
2 machine monitor on the computer hardware, booting the operating system on the virtual machine  
3 monitor, and devirtualizing the computer hardware before interposing the virtual machine  
4 monitor at runtime.
- 1 4. (Previously Presented) The method of claim 1, further comprising devirtualizing the  
2 computer hardware at runtime after the virtual machine monitor has been interposed.
- 1 5. (Original) The method of claim 1, wherein the computer hardware includes a CPU; and  
2 wherein the virtual machine monitor is interposed on the CPU.

1 6. (Currently Amended) The method of claim 5, wherein the computer hardware further  
2 includes memory, and the virtual machine monitor and the operating system each include CPU  
3 interrupt handlers; and wherein interposing the virtual machine monitor on the CPU includes:  
4 causing privileged instructions to trap to the virtual machine monitor, and  
5 redirecting interrupts to the corresponding virtual machine monitor CPU interrupt  
6 handlers instead of to the operating system CPU interrupt handlers.

1 7. (Original) The method of claim 6, wherein the privileged instructions are caused to trap  
2 to the virtual machine monitor by causing the operating system to run at a reduced privilege  
3 level; and wherein interposing the virtual machine monitor on the CPU further includes returning  
4 control to the operating system at the reduced privilege level.

1 8. (Previously Presented) The method of claim 6, wherein the privileged instructions are  
2 caused to trap to the virtual machine monitor by using a kernel module of the operating system to  
3 reduce a privilege level of the operating system from a higher privilege level.

1 9. (Previously Presented) The method of claim 6, wherein interposing the virtual machine  
2 monitor on the CPU further includes disabling physical memory access by the operating system.

1 10. ((Previously Presented) The method of claim 6, wherein interposing the virtual machine  
2 monitor on the CPU further includes loading the virtual machine monitor into the memory.

1 11. (Previously Presented) The method of claim 10, further comprising using a kernel  
2 module of the operating system to allocate memory within the operating system, pin the  
3 allocated memory, and load the virtual machine monitor into the pinned memory.

1 12. (Original) The method of claim 5, wherein the computer hardware includes memory; and  
2 wherein the virtual machine monitor is also interposed on the memory.

1 13. (Previously Presented) The method of claim 12, wherein interposing the virtual machine  
2 monitor on the memory includes partitioning the memory to provide partitions, and giving the  
3 virtual machine monitor access to at least one of the partitions.

1 14. (Original) The method of claim 12, wherein interposing the virtual machine monitor on  
2 the memory includes using a kernel module of the operating system to allocate a block of the  
3 memory, pin the block to prevent the operating system from using the block, and allocate the  
4 pinned block to the virtual machine monitor.

1 15. (Previously Presented) The method of claim 12, wherein interposing the virtual machine  
2 monitor on the memory includes commencing using the virtual machine monitor at runtime to  
3 manage memory translation.

1 16. (Original) The method of claim 5, wherein the computer hardware includes an I/O  
2 device, and wherein the virtual machine monitor is also interposed on the I/O device.

1 17. (Previously Presented) The method of claim 16, wherein the operating system includes a  
2 dual-mode driver that performs direct hardware control in a first mode and communicates with a  
3 device driver of the virtual machine monitor in a second mode; and wherein interposing the  
4 virtual machine monitor on the I/O device includes:  
5 setting the dual-mode driver to the second mode; and  
6 redirecting I/O interrupts to interrupt handlers in the virtual machine monitor instead of to  
7 interrupt handlers in the operating system.

1 18. (Previously Presented) The method of claim 16, wherein interposing the virtual machine  
2 monitor on the I/O device includes commencing I/O emulation of the I/O device at runtime.

1 19. (Previously Presented) A method of using a virtual machine monitor and an operating  
2 system on virtualized computer hardware, the method comprising devirtualizing the virtualized  
3 computer hardware at runtime of a computer containing the virtualized computer hardware,  
4 wherein runtime includes a period of execution in the computer after booting and before  
5 shutdown,  
6 wherein devirtualizing the virtualized computer hardware comprises stopping the virtual  
7 machine monitor.

1 20. (Previously Presented) The method of claim 19, wherein the virtualized computer  
2 hardware includes a CPU; and wherein the CPU is devirtualized at runtime.

1 21. (Currently Amended) The method of claim 20, wherein the virtualized computer  
2 hardware further includes physical memory, and the virtual machine monitor and the operating  
3 system each include CPU interrupt handlers; and wherein devirtualizing the CPU includes  
4 redirecting interrupts to the corresponding operating system CPU interrupt handlers instead of to  
5 the virtual machine monitor CPU interrupt handlers.

1 22. (Previously Presented) The method of claim 21, wherein devirtualizing the CPU further  
2 includes restoring a privilege level of the operating system from a less privileged mode to a more  
3 privileged mode.

1 23. (Previously Presented) The method of claim 21, wherein devirtualizing the CPU further  
2 includes enabling physical memory access by the operating system.

1 24. (Previously Presented) The method of claim 21, wherein devirtualizing the CPU further  
2 includes unloading the virtual machine monitor from the physical memory.

1 25. (Previously Presented) The method of claim 19, wherein the virtualized computer  
2 hardware includes memory; and wherein the memory is devirtualized at runtime.

1 26. (Original) The method of claim 25, wherein memory was allocated from the operating  
2 system to the virtual machine monitor during virtualization of the memory; and wherein  
3 devirtualizing the memory includes returning the allocated memory to the operating system.

1 27. (Previously Presented) The method of claim 25, wherein devirtualizing the memory  
2 includes remapping physical memory and using the operating system to manage address  
3 translation with respect to the devirtualized memory.

1 28. (Previously Presented) The method of claim 19, wherein the virtualized computer  
2 hardware includes an I/O device, and wherein the I/O device is devirtualized at runtime.

1 29. (Currently Amended) The method of claim 28, wherein the operating system includes a  
2 dual-mode driver that performs direct hardware control in a first mode and communicates with a  
3 device driver of the virtual machine monitor in a second mode; and wherein devirtualizing the  
4 I/O device includes:  
5 setting the dual-mode driver to the first mode from the second mode, and  
6 redirecting I/O interrupts to handlers in the operating system instead of [[the ]]handlers in  
7 the virtual machine monitor.

1 30. (Original) The method of claim 28, wherein devirtualizing the I/O device includes  
2 ceasing emulation of the I/O device at runtime.

1 31. (Previously Presented) A computer comprising hardware, the hardware including  
2 memory, the memory encoded with an operating system, a virtual machine monitor, and code for  
3 interposing the virtual machine monitor between the operating system and the hardware at  
4 runtime, wherein the interposing occurs after booting of the computer,  
5 wherein the operating system is to be booted in the computer before interposing the  
6 virtual machine monitor.

1 32. (Currently Amended) The computer of claim 31, wherein the hardware further includes a  
2 CPU, wherein the virtual machine monitor is interposed on the CPU at runtime, and the virtual  
3 machine monitor and the operating system each include CPU interrupt handlers; and wherein the  
4 interposing code is to cause privileged instructions to trap to the virtual machine monitor, and to  
5 redirect interrupts and traps to the corresponding virtual machine monitor CPU interrupt handlers  
6 instead of to the operating system CPU interrupt handlers.

1 33. (Previously Presented) The computer of claim 32, wherein the interposing code is to  
2 cause privileged instructions to trap to the virtual machine monitor by causing the operating  
3 system to run at a reduced privilege level from a higher privilege level; and wherein the  
4 interposing code is to reduce a privilege level of the operating system after redirecting the  
5 interrupts, and to return control to the operating system at the reduced privilege level.

1 34. (Previously Presented) The computer of claim 32, wherein the interposing code includes  
2 a kernel module of the operating system for reducing a privilege level of the operating system  
3 from a higher privilege level, whereby the privileged instructions trap to the virtual machine  
4 monitor.

1 35. (Previously Presented) The computer of claim 32, wherein the interposing code is to  
2 disable physical memory access by the operating system.

1 36. (Previously Presented) The computer of claim 31, wherein the interposing code includes  
2 a kernel module of the operating system for allocating a block of the memory, pinning the block  
3 to prevent the operating system from using the block, and allocating the pinned block to the  
4 virtual machine monitor, whereby the virtual machine monitor is interposed on the memory at  
5 runtime.

1 37. (Previously Presented) The computer claim 31, wherein the interposing code is to  
2 commence using the virtual machine monitor at runtime to manage memory translation, whereby  
3 the virtual machine monitor is interposed on the memory at runtime.

1 38. (Previously Presented) The computer of claim 31, wherein the hardware further includes  
2 an I/O device; and wherein the interposing code includes an operating system dual-mode driver  
3 to perform direct hardware control in a first mode and to communicate with a device driver of  
4 the virtual machine monitor in a second mode; and wherein the interposing code is to set the  
5 dual-mode driver to the second mode, and to direct I/O interrupts to interrupt handlers in the  
6 virtual machine monitor instead of to interrupt handlers in the operating system, whereby the  
7 virtual machine monitor is interposed on the I/O device at runtime.

1 39. (Previously Presented) The computer of claim 31, wherein the hardware further includes  
2 an I/O device; and wherein the operating system includes a dual-mode driver to perform direct  
3 hardware control in a first mode and to communicate with a device driver of the virtual machine  
4 monitor in a second mode; and wherein the interposing code is to set the dual-mode driver to the  
5 second mode, and to redirect I/O interrupts to interrupt handlers in the virtual machine monitor  
6 instead of to interrupt handlers in the operating system, whereby the virtual machine monitor is  
7 interposed on the I/O device.

1 40. (Previously Presented) The computer of claim 31, wherein the hardware further includes  
2 an I/O device; and wherein the interposing code is to commence I/O emulation of the I/O device  
3 at runtime, whereby the virtual machine monitor is interposed on the I/O device at runtime.

1 41. (Previously Presented) A computer comprising hardware, the hardware including  
2 memory, the memory encoded with a virtual machine monitor to virtualize the hardware, and  
3 code for devirtualizing the hardware at runtime, wherein runtime includes a period of execution  
4 in the computer after booting and before shutdown, and wherein devirtualizing the hardware  
5 comprises stopping the virtual machine monitor.

1 42. (Previously Presented) The computer of claim 41, wherein the hardware further includes  
2 a CPU; and wherein the devirtualizing code is to devirtualize the CPU at runtime.

1 43. (Previously Presented) The computer of claim 42, wherein the memory is further encoded  
2 with an operating system including interrupt handlers; wherein the virtual machine monitor

3 includes interrupt handlers; and wherein the devirtualizing code is to redirect interrupts to the  
4 corresponding interrupt handlers of the operating system instead of to the interrupt handlers of  
5 the virtual machine monitor.

1 44. (Previously Presented) The computer of claim 43, wherein the devirtualizing code is to  
2 restore privilege level of the operating system from a lower privilege level to a higher privilege  
3 level.

1 45. (Previously Presented) The computer of claim 43, wherein the devirtualizing code is to  
2 enable physical memory access by the operating system.

1 46. (Previously Presented) The computer of claim 41, wherein the devirtualizing code is to  
2 devirtualize the memory at runtime.

1 47. (Previously Presented) The computer of claim 46, wherein the virtual machine monitor is  
2 to allocate memory from an operating system to the virtual machine monitor; and wherein the  
3 devirtualizing code is to return the allocated memory to the operating system.

1 48. (Cancelled)

1 49. (Previously Presented) The computer of claim 41, wherein the hardware includes an I/O  
2 device, wherein the virtual machine monitor is to virtualize the I/O device; and wherein the  
3 devirtualizing code is to devirtualize the I/O device at runtime.

1 50. (Previously Presented) The computer of claim 49, wherein the memory is further encoded  
2 with an operating system including dual-mode drivers to perform direct hardware control in a  
3 first mode and communicate with device drivers of the virtual machine monitor in a second  
4 mode; and wherein the devirtualizing code is to set the dual-mode drivers to the first mode from  
5 the second mode, and to redirect I/O interrupts to handlers in the operating system instead of to  
6 handlers in the virtual machine monitor.



- 1 51. (Previously Presented) The computer of claim 49, wherein the devirtualizing code is to  
2 cease emulation of the I/O device at runtime.
- 1 52. (Previously Presented) An article for use with an operating system on computer  
2 hardware, the article comprising a computer-readable storage medium storing software that when  
3 executed by the computer causes the computer to:  
4 virtualize at least a portion of the computer hardware at runtime by providing a virtual  
5 machine monitor between the operating system and the computer hardware, wherein the  
6 virtualizing occurs after booting of the computer and loading of the operating system, and  
7 wherein the operating system is to be booted in the computer before virtualizing the at  
8 least a portion of the computer hardware at runtime.
- 1 53. (Previously Presented) The article of claim 52, wherein the computer hardware further  
2 includes a CPU, and wherein the virtual machine monitor and the operating system each include  
3 CPU interrupt handlers; and wherein the software is executable to cause privileged instructions  
4 to trap to the virtual machine monitor, and to cause interrupts and traps to be redirected to the  
5 corresponding virtual machine monitor interrupt handlers instead of to the operating system  
6 interrupt handlers.
- 1 54. (Previously Presented) The article of claim 53, wherein the software is executable to  
2 cause the privileged instructions to trap to the virtual machine monitor by reducing a privilege  
3 level of the operating system from a higher privilege level, and wherein the software causes  
4 control to be returned to the operating system at the reduced privilege level.
- 1 55. (Previously Presented) The article of claim 53, wherein the software is executable to  
2 cause physical memory access by the operating system to be disabled.
- 1 56. (Previously Presented) The article of claim 52, wherein the computer hardware includes  
2 memory, and wherein the virtual machine monitor is for causing a kernel module of the  
3 operating system to allocate a block of the memory, pin the block to prevent the operating  
4 system from using the block, and allocate the pinned block to the virtual machine monitor.

1 57. (Cancelled)

1 58. (Currently Amended) The article of claim 52, wherein the computer hardware further  
2 includes an I/O device; and wherein the software includes an operating system dual-mode driver  
3 to perform direct hardware control in a first mode and communicate with a corresponding device  
4 driver of a virtual machine monitor in a second mode; and wherein the dual-mode driver is set to  
5 the second mode when the at least the portion of the computer hardware is virtualized, and  
6 wherein I/O interrupts are redirected to interrupt handlers in the virtual machine monitor instead  
7 of [[the ]]interrupt handlers in the operating system.

1 59. (Previously Presented) The article of claim 52, wherein the computer hardware further  
2 includes an I/O device; and wherein the operating system includes a dual-mode driver to perform  
3 direct hardware control in a first mode and communicate with a device driver of the virtual  
4 machine monitor in a second mode; and wherein the dual-mode driver is set to the second mode  
5 when the at least the portion of the computer hardware is virtualized, and wherein I/O interrupts  
6 are redirected from interrupt handlers in the operating system to interrupt handlers in the virtual  
7 machine monitor.

1 60. (Previously Presented) The article of claim 52, wherein the computer hardware further  
2 includes an I/O device; and wherein the software is executable to cause I/O emulation of the I/O  
3 device to commence at runtime.

1 61. (Original) An article for running an operating system and a virtual machine monitor on a  
2 computer, the computer including an I/O device, the article comprising computer memory  
3 encoded with an I/O driver having first and second modes of operation, the I/O driver operable in  
4 the first mode to interface directly between the operating system and the I/O device, the I/O  
5 driver operable in the second mode to interface between the operating system and a  
6 corresponding I/O driver of the virtual machine monitor.

1 62. (Previously Presented) An article for use with an operating system on computer  
2 hardware, the article comprising a computer-readable storage medium storing software that when  
3 executed by a computer causes the computer to devirtualize at least a portion of virtualized  
4 hardware at runtime, wherein runtime is a period of execution in the computer after booting and  
5 before shutdown, and wherein devirtualizing the at least a portion of the virtualized hardware  
6 comprises stopping a virtual machine monitor interposed between the operating system and the  
7 hardware.

1 63. (Previously Presented) The article of claim 62, wherein the virtualized hardware includes  
2 a CPU; and wherein the software causes the CPU to be devirtualized at runtime.

1 64. (Previously Presented) The article of claim 63, wherein the virtualized hardware further  
2 includes memory, and wherein the memory is further encoded with the operating system  
3 including first interrupt handlers; wherein the software includes second interrupt handlers; and  
4 wherein the software is executable to cause interrupts to be redirected to the corresponding first  
5 interrupt handlers instead of to the second interrupt handlers.

1 65. (Previously Presented) The article of claim 64, wherein the software is executable to  
2 cause a privilege level of the operating system to be restored from a lower privilege level to a  
3 higher privilege level.

1 66. (Previously Presented) The article of claim 64, wherein the software is executable to  
2 cause physical memory access by the operating system to be enabled.

1 67. (Previously Presented) The article of claim 62, wherein the virtualized hardware includes  
2 a memory, and wherein the software is executable to cause the memory to be devirtualized at  
3 runtime.

1 68. (Currently Amended) The article of claim 67, wherein if a part of the memory was  
2 allocated from an operating system to the virtual machine monitor prior to the runtime  
3 devirtualization, the software is executable to cause ~~causes~~ the allocated memory to be returned  
4 to the operating system as part of the runtime devirtualization.

1 69. (Previously Presented) The article of claim 67, wherein the software is executable to  
2 cause physical memory to be remapped and wherein the software allows an operating system to  
3 manage address translation with respect to the devirtualized memory.

1 70. (Previously Presented) The article of claim 62, wherein the virtualized hardware includes  
2 an I/O device; and wherein the software is executable to cause the I/O device to be devirtualized  
3 at runtime.

1 71. (Previously Presented) The article of claim 70, wherein the virtualized hardware further  
2 includes a memory, and wherein the memory is further encoded with the operating system  
3 including dual-mode drivers that perform direct hardware control in a first mode and  
4 communicate with virtual device drivers in a second mode; and wherein the software is  
5 executable to cause the dual-mode drivers to be set to the first mode.

1 72. (Previously Presented) The article of claim 70, wherein the software is executable to  
2 cause emulation of the I/O device to cease at runtime.

1 73. (Previously Presented) The computer of claim 31, wherein interposing the virtual  
2 machine monitor gives the virtual machine monitor direct control of at least a portion of the

3 hardware such that the operating system no longer has direct control of the at least a portion of  
4 the hardware.

1 74. (Previously Presented) The article of claim 52, wherein providing the virtual machine  
2 monitor between the operating system and the computer hardware gives the virtual machine  
3 monitor direct control of at least a portion of the hardware such that the operating system no  
4 longer has direct control of the at least a portion of the hardware.